

ECE 471 – Embedded Systems

Lecture 14

Vince Weaver

`http://web.eece.maine.edu/~vweaver`

`vincent.weaver@maine.edu`

29 September 2017

Announcements

- Homework #4 was due.
- Homework #5 will be posted today, due in two weeks.
- Hand out i2c displays today. Be careful with them!
- Midterm coming up next week in Wednesday class.



i2c

- Inter-Integrated Circuit, Invented by Philips (now NXP) in 1982
- Broadcom and others for some reason call it “Two Wire Interface”
- Two-wires (4 if you include Vdd and Ground)
- Since 2006, no licensing fees (though do have to pay to reserve number)



Why is i2c popular?

- Stable standard
- Relatively easy to implement
- Not many wires
- Good enough
- Cheap



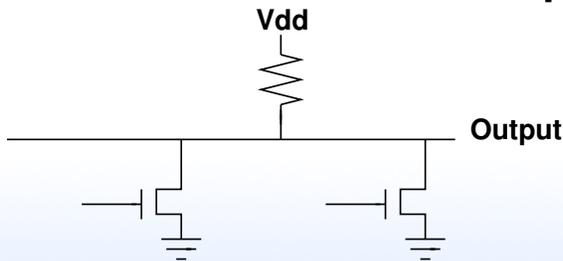
Uses of i2c

- SMBus
- DDC (VGA/HDMI) (video card / monitor communication)
- Configuring SDRAM
- Temp sensor and fan chips on motherboards
- Wii nunchuck



Protocol Overview

- Serial Data Line (SDA) and Serial Clock (SCL), Open Drain, Pulled up by resistors
- Open drain means output can be wired together
If not driven, high-Z, line floats high
If driven, pulls to zero
Can have multiple connected to one line, “wired-or”



- 7-bit (or 10-bit) address
- Standard=100kbits/s, slow=10kbits/s, fast=400kbits/s
fast plus 1Mbits/s, high 3.4Mbits/s (actual transfers
slower due to overhead)
- Length of bus limited to a few meters (400pF)



High-level Protocol

- Master (generates clock, init transaction), Slaves (responds)
- Can be multiple masters and slaves
- Master sends start bit, 7-bit address of slave, then read/write bit
- Slave responds with ACK
- Reads and writes are 8 bits of data, followed by 1 ACK



bit

- Send stop bit when done
- Address and Data set Most-significant Bit first



Low-level Protocol

- Busses start out floating high (by pull-up resistors)
- Start bit: SDA transition high-low while SCL high
- To transmit bit, master sets SCL low, then sets SDA to value, lets SCL float high, wait 4us, set SCL low for next cycle
- Clock stretching: slave can hold SCL low until it is done processing, master should check to be sure SCL floated back high before continuing.
- After every 8-bits other side sends ACK bit. If 0, more



to come. If 1, we are done (or there is an error)
ACK the master toggles clock then reads the SDA line.
This is also how a full 8-bit read happens

- Stop bit: SDA transition low-high while SCL high (only start/stop SDA transitions happen when SCL is high).
- Arbitration: masters monitor SDA and won't start unless idle. Deterministic arbitration.
If tries to send a 1 and notices something else is pulling to zero, then a collision and stops. Low addresses



automatically win.

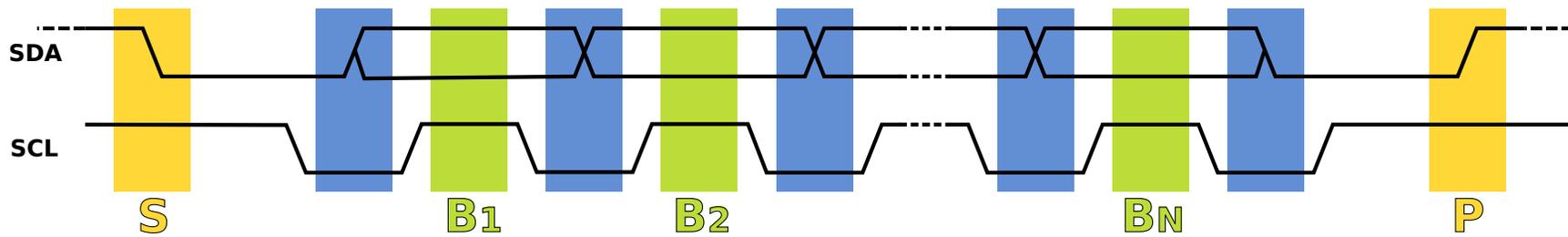


Figure 1: Protocol diagram from Wikipedia



Message Types

- Master writing to slave:
Sends start, address, write (0), waits for ACK (low), then sends 8 bits of data, waits for ACK, etc.
- Master reading from slave:
Sends start, address, read (1), waits for ACK (low), then waits for 8 bits, sends ACK if wants more, otherwise stop if done.
- Combined – can send multiple messages or to multiple



slaves by not sending stop but and instead sending a new start bit



i2c Reserved Addresses

Address	R/W Bit	Description
000 0000	0	General call address
000 0000	1	START byte
000 0001	X	CBUS address
000 0010	X	Reserved for different bus format
000 0011	X	Reserved for future purposes
000 01XX	X	Hs-mode master code
111 10XX	X	10-bit slave addressing
111 11XX	X	Reserved for future purposes

10-bit addresses work by using special address above with first 2 bits + R/W, then sending an additional byte with the lower 8 bits.



SMbus

- Enhanced i2c bus interface
- Has stricter rules about some signals
- Can do more advanced things, such as have slaves send notifications to master



i2c and Rasp-pi

- 3+ busses, only one easily accessible
on Model B and newer one on camera interface
on Model 2B/3B one for “hat” EEPROM
on Model 3B GPIO extender, driven by GPU?
- `modprobe i2c-bcm2708` and `i2c-dev`
May also want to edit `/etc/modules` and remove from
blacklist `/etc/modprobe.d/raspi-blacklist.conf`
- Also want to install `i2c-tools` if possible `apt-get`
`i2c-tools`



- i2c port 1 (/dev/i2c-1). Used to be i2c-0 on older machines. Other boards (beaglebone black) likely different.
- Clock stretching buggy on the rasp-pi
- 3.3V
- default speed is 100kHz. You can change this with the baudrate= module parameter.
- i2c-1 on pins SDA=3, SCL=5
- i2c-0 on the camera interface (pad5)



i2c and Linux

- Like with GPIOs, kernel can drive it, or be exposed to userspace
- i2c-dev module must be installed (and i2c driver)
- Open the device node, `/dev/i2c-1`
- Use ioctls `I2C_SLAVE` to set the address of the device we wish to talk to.
- Use standard read or write calls to communicate with



the device

- Close the device when done.
- i2c slave addresses are 7 bits, but when sent the r/w bit is put at end. This can be confusing; some spec sheets will list a slave address as 0xE0/0xE1 (8 bits, including r/w) but Linux exports this as 0x70 (0xE0 shifted right by 1).



Sample i2c Linux code

For more details on this, see the HW#5 handout.

```
unsigned char buffer[17];
int display_fd;

/* open */
display_fd = open("/dev/i2c-1", O_RDWR);
if (display_fd < 0) fprintf(stderr, "Error!\n");

/* set slave address */
result=ioctl(display_fd, I2C_SLAVE, 0x70);
if (result < 0) fprintf(stderr, "Error!\n");

/* writing */
buffer[0]= HT16K33_REGISTER_SYSTEM_SETUP | 0x01;
```



```
if ( (write(display_fd , buffer , 1)) !=1) {  
    fprintf(stderr , " Error!\n" );  
}
```

```
/* closing */  
close( display_fd );
```



i2c on the Pi – detecting

```
i2cdetect -y -r 1
```

```
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: 70 -- -- -- -- -- -- --
```



LED Driver Chip

- This is a ht16k33, datasheet available:

<http://www.adafruit.com/datasheets/ht16K33v110.pdf>

- Supports up to 16x8 LEDs, as well as keypad input. Can dim display, also blink. Common cathode.

-|>|- common

- Works by rapidly scanning all segments fast enough cannot see.



- To set up, write byte commands, high 4 bits command lower 4 bits data.
- To set up full display, write the pointer offset of internal framebuffer, than 16 bytes of on/off data.
- Actual LED hooked up is a BL-Q56D-43UG 4x7 segment Ultra-Green display common cathode.
- How do you set address? (have more than one display hooked up?)



Benefit of OS

- Code is portable across all machines with i2c bus
- Can use same code on Gumstix, Rasp-Pi, Beaglebone, etc.
- Will probably need to change the bus number (It's i2c-3 on gumstix).

